Neutrino Event Classification

N. Saoulidou and G. Tzanakos
University of Athens, Department of Physics
Donut Collaboration meeting: July 7, 2000

Outline

- Goals
- Method: Artificial Neural Networks (ANN)
- Monte Carlo Event Generation
- MC-Data Comparison
- Selection of Variables
- Preliminary Results
- Ongoing work

Goals

• Use classification techniques to classify/identify Neutrino Interactions on "event-by-event" basis using topological and physical characteristics of neutrino events derived from MC generated interactions:

• CC ν_μ ν_e ν_τ

• NC

• **Requirement:** MC should be capable of describing very well the neutrino data.

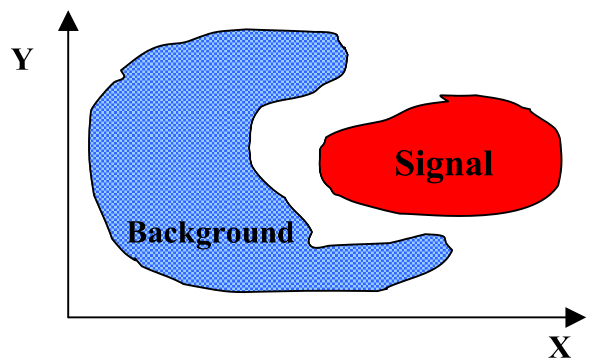
• **Classification Methods:** Method of Discriminants

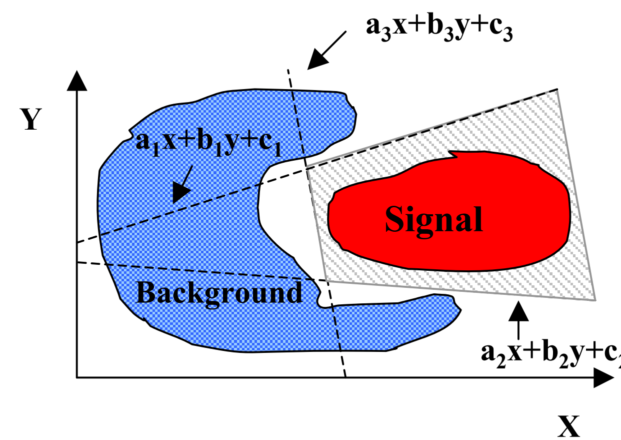
Artificial Neural Networks

Methods: Artificial Neural Networks

- ANN can be trained by MC generated events
- A trained ANN provides multidimensional cuts for data that are difficult to deduce in the usual manner from 1-d or 2-d histogram plots.
- ANN has been used in HEP
- HEP Packages:
  - JETNET
  - SNNS
  - MLP fit

ANN BASICS



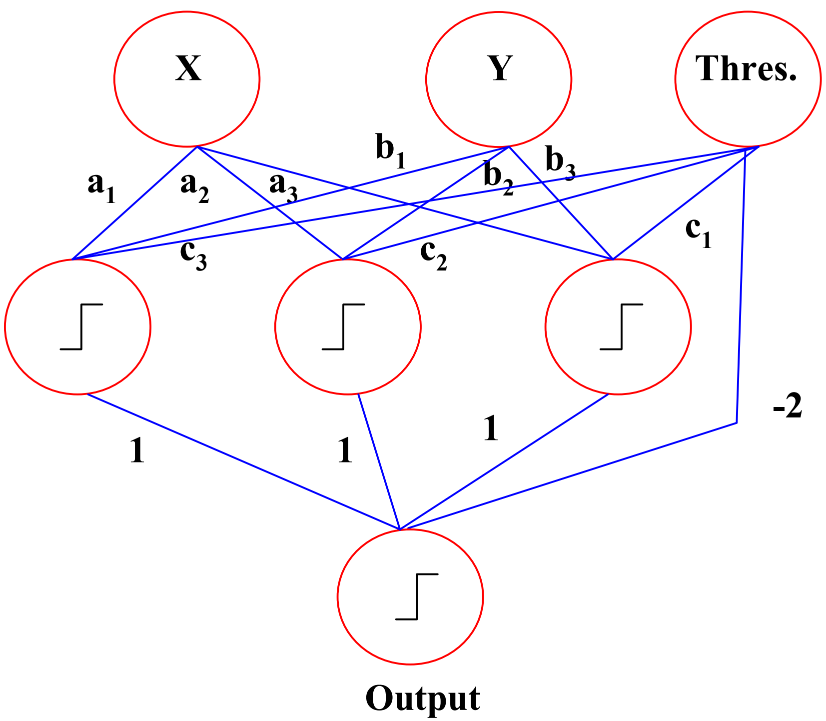


- Event sample characterized by two variables X and Y (left figure)
- A linear combination of cuts can separate "signal" from "background" (right fig.)
- Define "step function" $\mathbf{S(ax + by + c)} = \begin{cases} 0 & \text{"Signal (x, y)" OUT} \\ 1 & \text{"Signal (x, y)" IN} \end{cases}$
- Separate "signal" from "background" with the following function:

$$\mathbf{C(x,y) = S(S(a_1x + b_1y + c_1) + S(a_2x + b_2y + c_2) + S(a_3x + b_3y + c_3) - 2)}$$

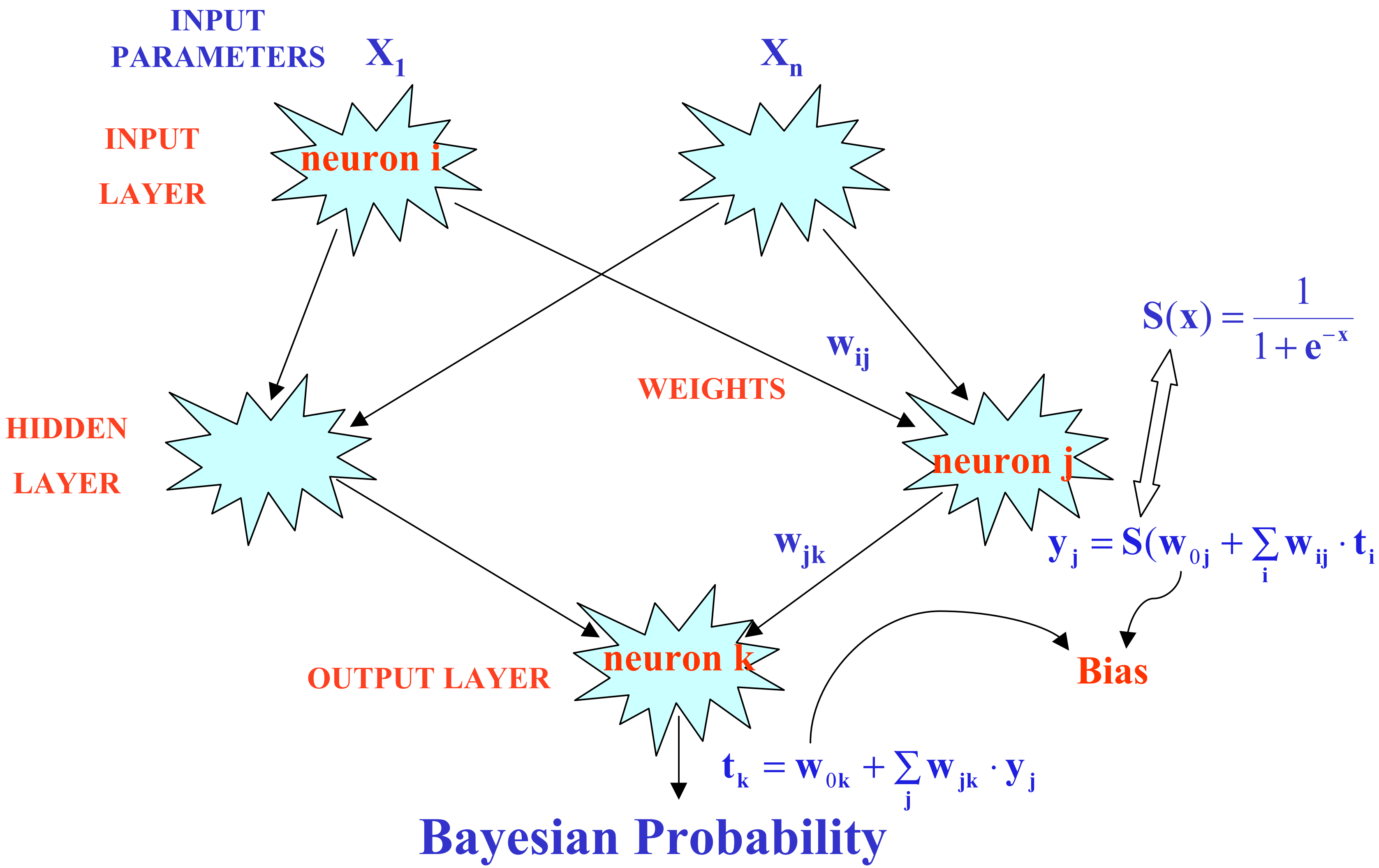
ANN BASICS

Visualization of function C(x,y)



- The diagram resembles a **feed forward neural network** with **two input neurons, three neurons** in the first **hidden layer** and **one output neuron**.
- **Threshold** produces the desired **offset**.
- Constants a_i, b_i are the **weights** $w_{i,j}$ (i and j are the neuron indices).

ANN Basics



INPUT PARAMETERS
X_1
X_n
INPUT LAYER
neuron i
HIDDEN LAYER
WEIGHTS
w_{ij}
neuron j
$S(x) = \frac{1}{1+e^{-x}}$
$y_j = S(w_{0j} + \sum_i w_{ij} \cdot t_i)$
w_{jk}
Bias
OUTPUT LAYER
neuron k
$t_k = w_{0k} + \sum_j w_{jk} \cdot y_j$
Bayesian Probability

ANN BASICS

- **Output** of t_j each neuron in the first hidden layer :

$$t_j = S(\sum_i w_{ij} \cdot t_i)$$

- **Transfer function** is the sigmoid function :

$$S(x) = \frac{1}{1+e^{-x}}$$

  - **For the standard backpropagation training procedure of neural networks, the derivative of the neuron transfer functions must exist in order to be able to minimize the network error (cost) function E.**
  - **Any continuous function of any number of variables on a compact set can be approximated to any accuracy by a linear combination of sigmoids**
  - **Trained with desired output 1 for signal and 0 for background the neural network function (output function t_j) approximates the Bayesian Probability of an event being a signal.**

ANN BASICS

- Error function : $\mathbf{E} = \sum_{p} \mathbf{E}_p = \sum_{jp} (\mathbf{d}_{pj} - \mathbf{t}_{pj})^2$, where
  - p : runs over the events of the training set,
  - j : the index of an output neuron,
  - d_{pj} : the desired output of neuron j in event p,
  - t_{pj} : the network output.

- All **minimization** methods use the computation of first order derivatives:

$$\frac{\partial \mathbf{E}}{\partial \mathbf{w}_{ji}} = \sum_{p} \frac{\partial \mathbf{E}_p}{\partial \mathbf{w}_{ji}}$$

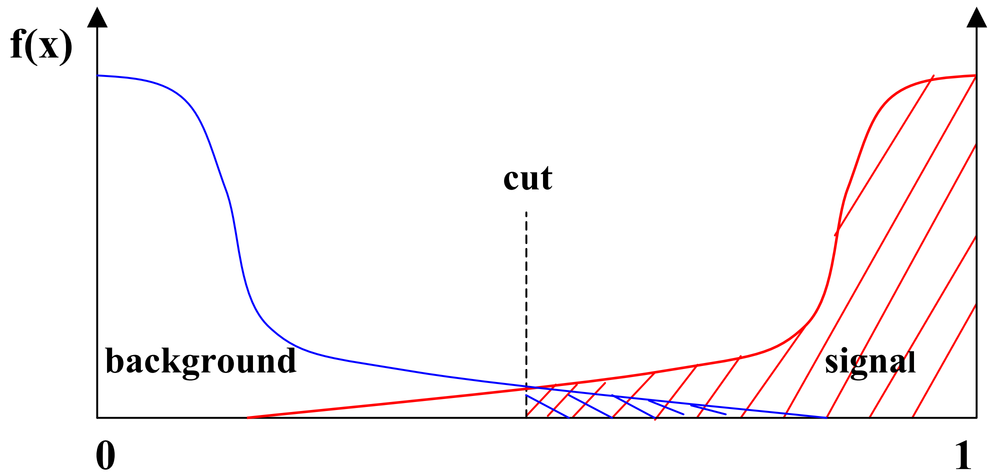
- The description of **backpropagation** is that in each iteration :

$$\Delta_p \mathbf{w}_{ji}(\mathbf{n}+1) = -\varepsilon \frac{\partial \mathbf{E}_P}{\partial \mathbf{w}_{ji}} + \mathbf{a}\Delta_p \mathbf{w}_{ji}(\mathbf{n})$$, where

  - $\Delta_p w_{ji}(n+1)$: the **change in w_{ji}** in iteration n+1,
  - ε : the distance to move along the gradient (‘**learning coefficient**’)
  - α : a smoothing term (“**momentum** ”)

ANN BASICS

Network output (selection) function for "background "and "signal" events



- **Signal Selection Efficiency :** $\text{efficiency} = \dfrac{Nsig_{cut}}{Nsig}$
  - Number of signal events above the cut / Total number of signal events
- **Signal Selection Purity :** $\text{purity} = \dfrac{Nsig_{cut}}{Nsig_{cut} + Nback_{cut}}$
  - Number of signal events above the cut / number of signal events above the cut plus the number of background events above the cut.

Monte Carlo Event Generation

- For the neural network training set we produced MC files with the following characteristics :

(A) Scintillating Fiber System

→ Scintillating Fiber Hits produced with **SF decoder 2**. When that analysis was performed we believed that SF2 decoder was giving better results. But that is not the case…→

→ **SF decoder 2** gives way to many hits in the Scintillating Fibers and the **tracking code fails** (Bruce Baller). So... →

→ At the end the **ANN analysis** has to be formed using **SF decoder 1** or a **modified version** of this since... →

→ **None** of this **decoders** (as will be seen later) **describes the data** in an acceptable way.

Monte Carlo Event Generation

(B) MC info & Smeared MC info

- The event distributions we use are produced using **Smeared MC hits** and not Ideal **MC hits**.
- **Smeared MC** hits should represent real hits since they are formed from MC hits with **convolution of errors**.
- We use the **MC weights** to weight our distributions in order to take into account the **probability** of an **event** to occur.

MC-Data Comparison

- **Method** : **Kolmogorov test**
- **Definition** : **"maximum value of the absolute difference between two cumulative distribution functions".**
- **Equation** : $D = \max_{-\infty < x < \infty} \left| S_{N_1}(x) - S_{N_2}(x) \right|$, where $S_{N_1}(x)$ and $S_{N_2}(x)$ are the cumulative distribution functions.
- **Statistical principle** : **Distinguish** between the **null hypothesis** (the two distributions - histograms) are compatible and the **alternative hypothesis**.
- **PAW - HBOOK implementation** : routine **HDIFF**
- **Output** : the **probability** of **compatibility** between **two histograms** (the two histograms coming from the same parent distribution).
- **Probability Criterion** : **Common choices** are **0.05 0.01 0.001**. That is if ones accept that **2 histograms** are **compatible** whenever the **probability** output of the Kolmogorov test is **greater than 0.05** then **truly compatible histograms** should **fail the test** exactly **5 % of the time.**

MC-Data Comparison

Variables studied

nsfhitrec = Total number of Scintillating Fiber hits

pulse_hgt = Total pulse_hgt of Scintillating fibers

ntksf = total number of SF tracks

nsfh_st1 = Percentage of hits in the first Station

nsfh_st2 = Percentage of hits in the second Station

nsfh_st3 = Percentage of hits in the third Station

nsfh_st4 = Percentage of hits in the fourth Station

ndchitrec = Total number of Drift Chamber Hits

ntkdc = Total number of Drift Chamber Tracks

emtotreco = Total Energy Deposition in the EMCAL

nclu = Number of clusters in the EMCAL

avene = Average Cluster Energy in the EMCAL

nmidhitrec = Total Number of hits in the Muon Identification System (MID)

nmidhitrec_sc = Total Number of hits in the Scintillating Tubes (MID)

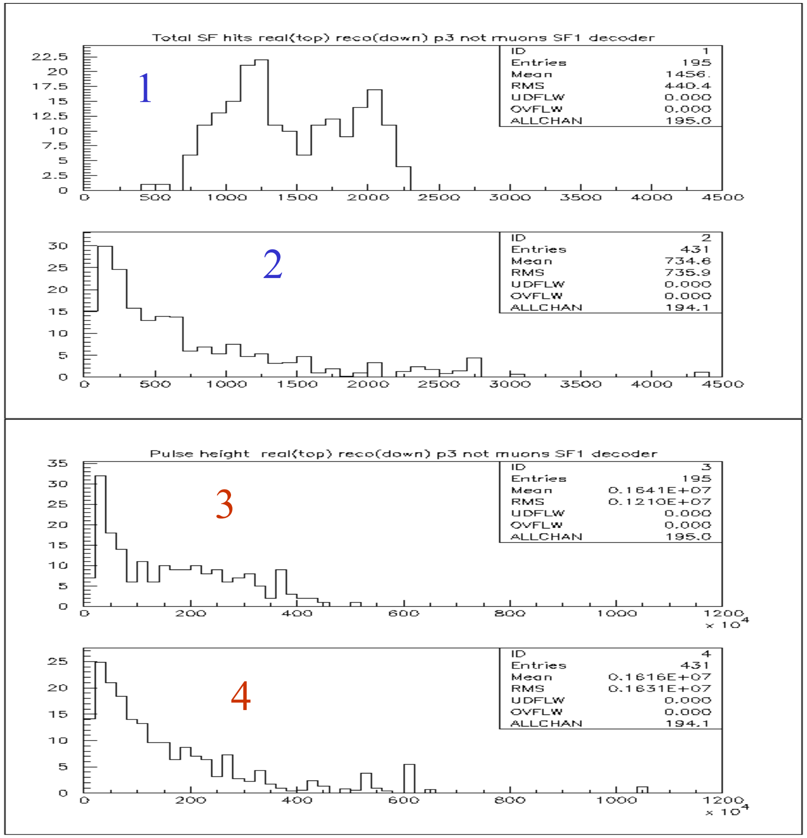
nmd_st1 = Percentage of MID hits in Wall A

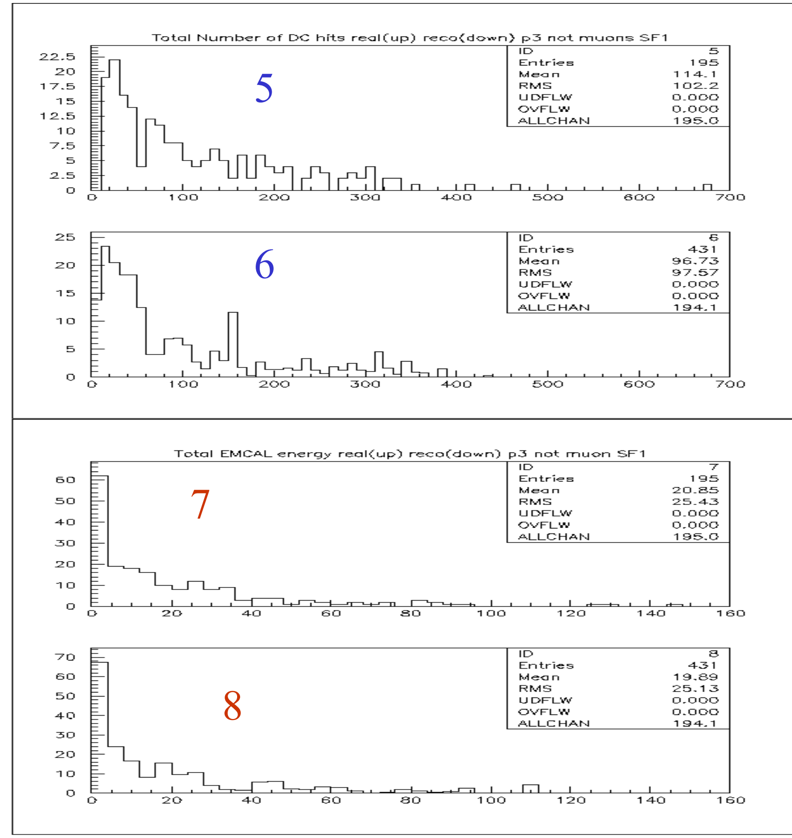
nmd_st2 = Percentage of MID hits in Wall B

nmd_st3 = Percentage of MID hits in Wall C

ntkfin = Total number of “final” tracks

MC-Data Comparison (Cont)





1: **SF hits (data)**

2: SF hits (MC, reco)

3: Pulse height (data)

4: Pulse height (MC, reco)

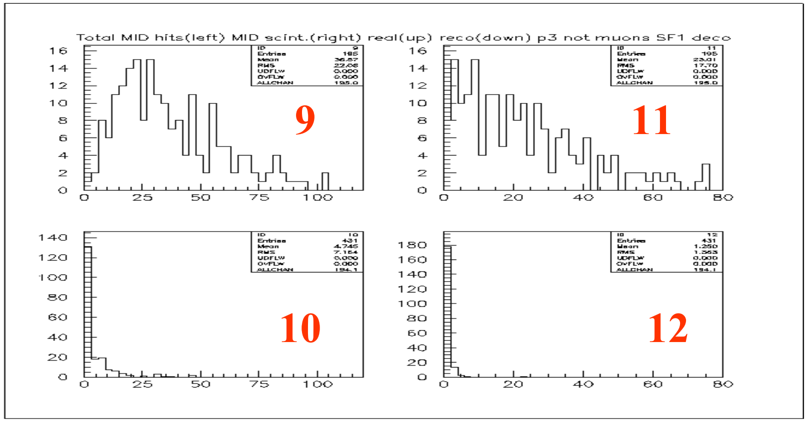
5: DC hits (data)

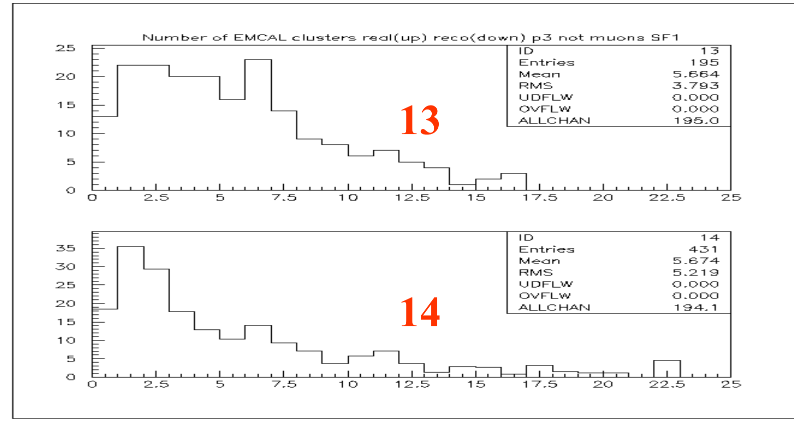
6: DC hits (MC, reco)

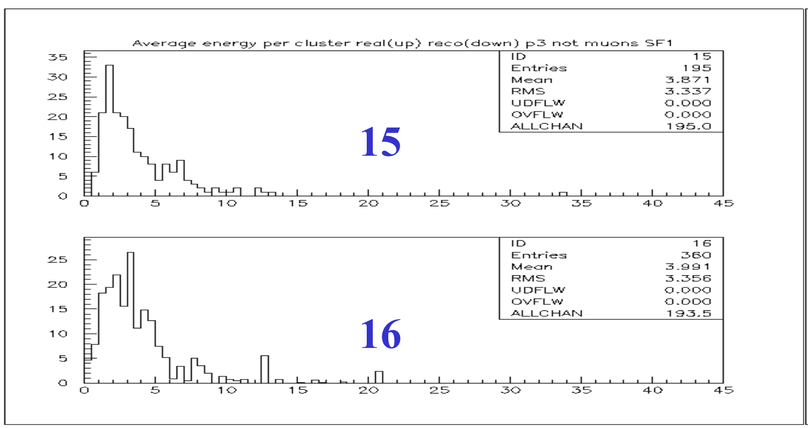
7: EMCAL energy (data)

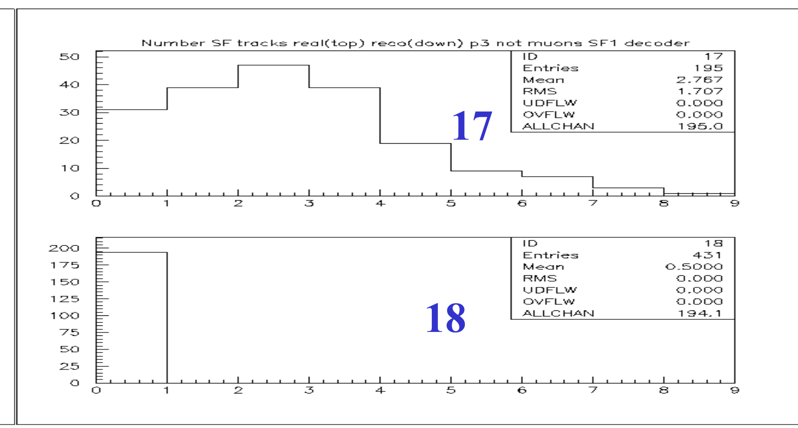
8: EMCAL Energy (MC, reco)

MC-Data Comparison









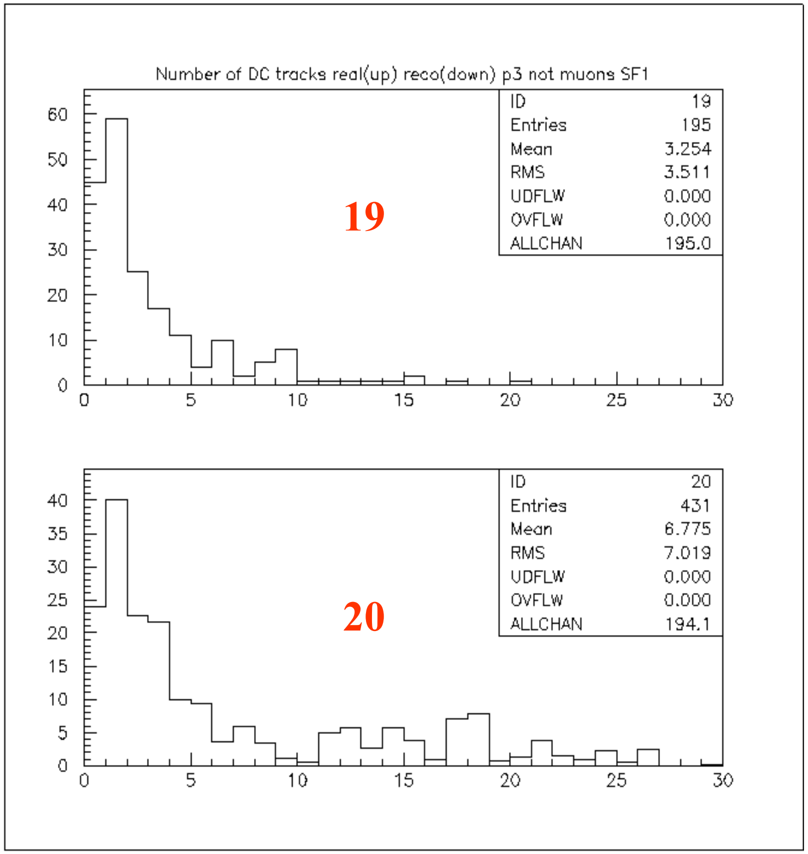
9: MID hits (data) **11: MID scint. Hits (data)** **13: Number of clusters (data)**

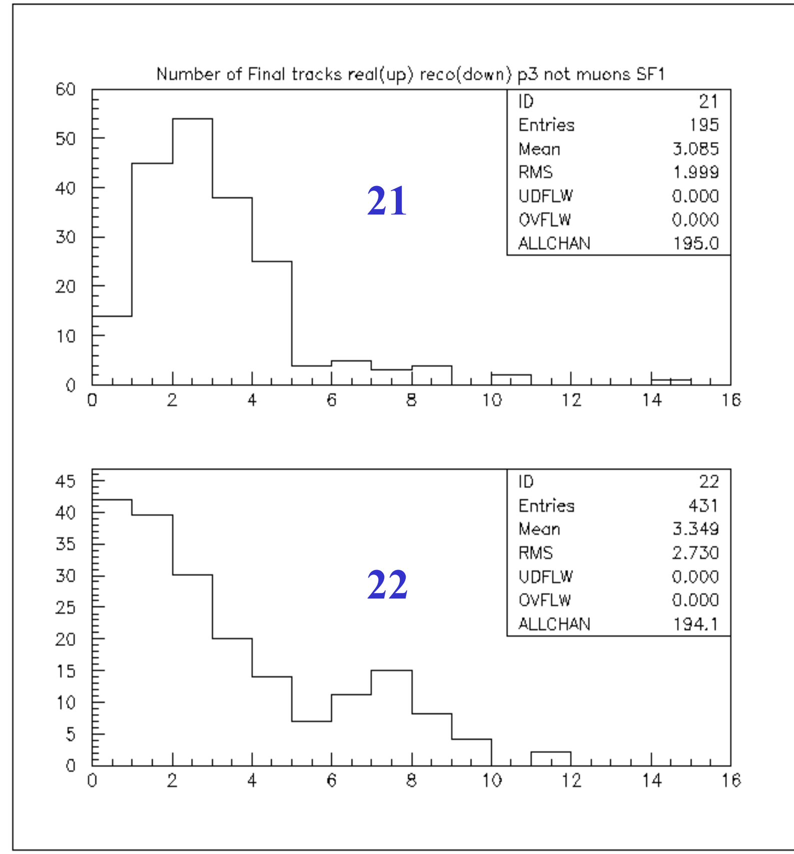
10: MID hits (MC, reco) **12: MID scint. Hits (MC, reco)** **14: Number of clusters (MC, reco)**

15: Average Cluster energy (data) **17: Number of SF tracks (data)**

16: Average Cluster energy (MC, reco) **18: Number of SF tracks (MC, reco)**

MC-Data Comparison





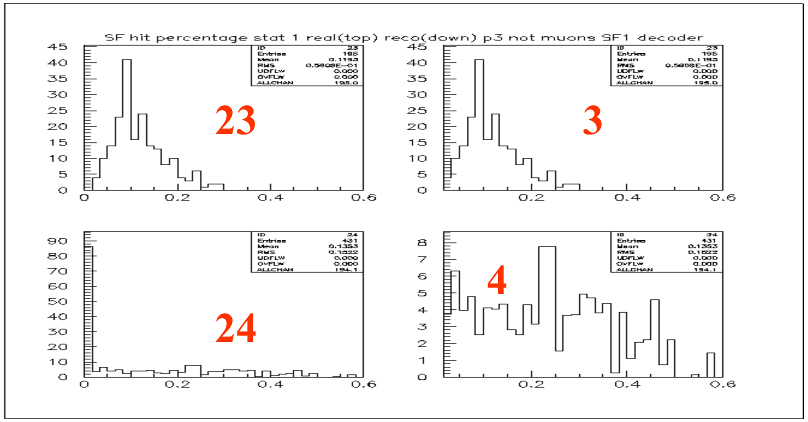
19: Number of DC tracks (data)

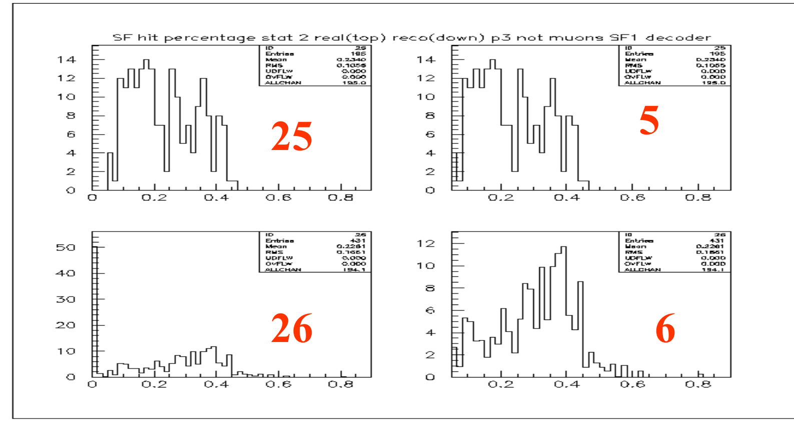
20: Number of DC tracks (MC, reco)

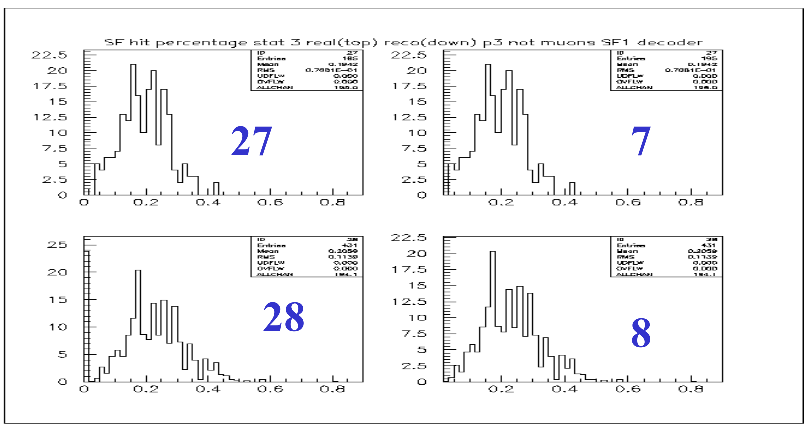
21: Number of "final" tracks (data)

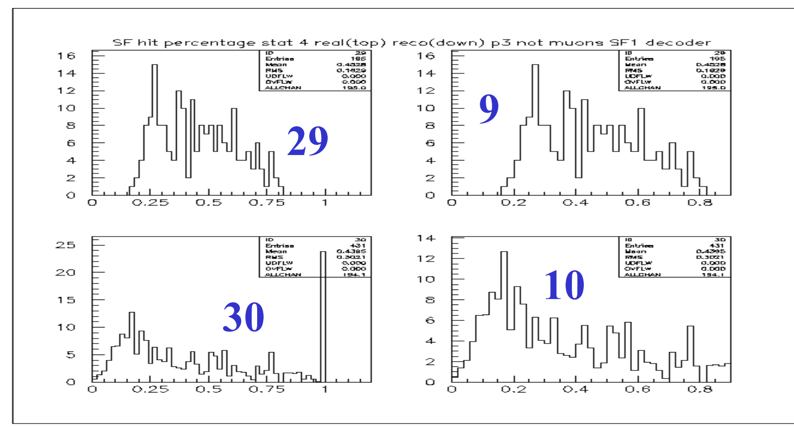
22: Number of "final" tracks (MC, reco)

MC-Data Comparison









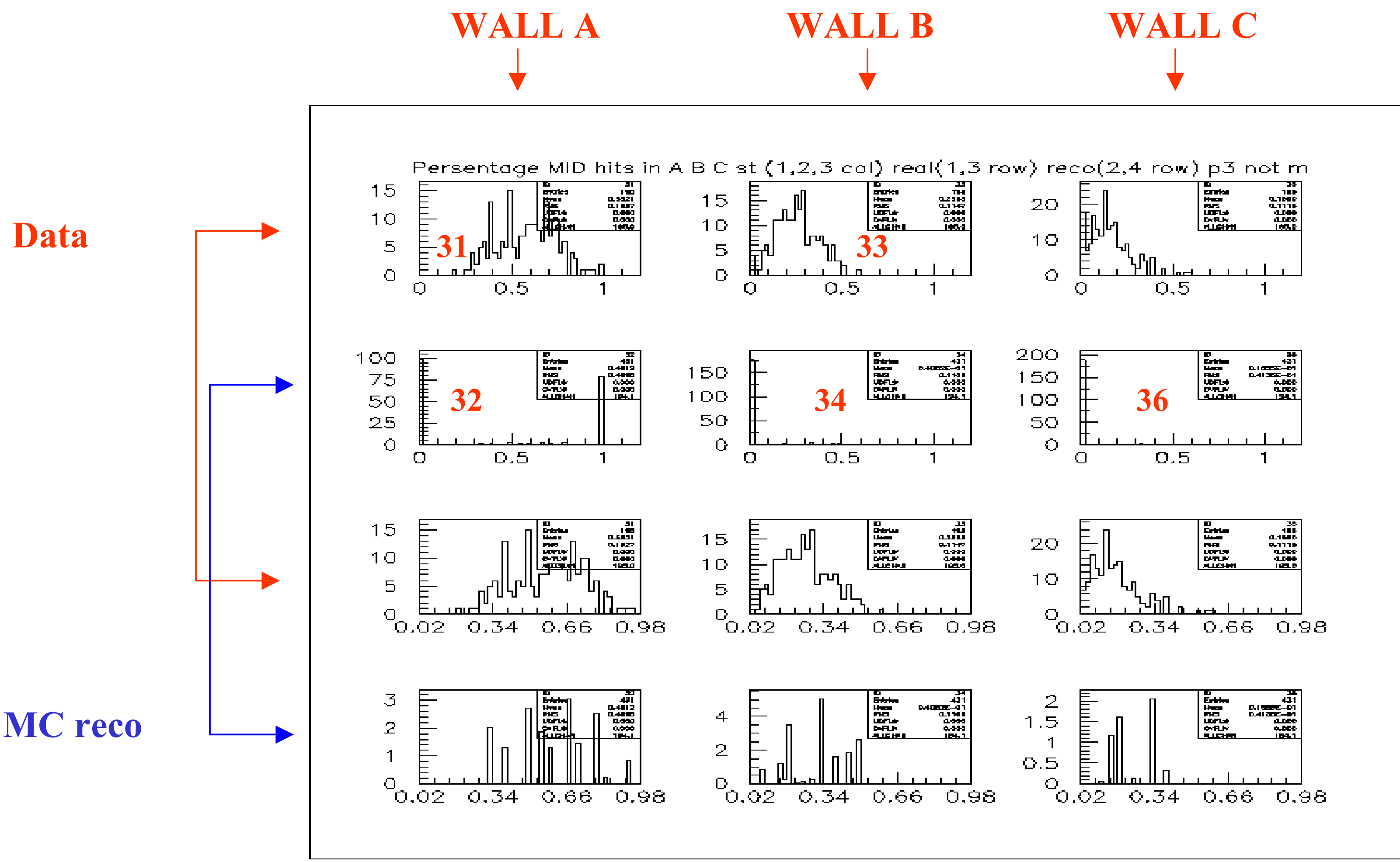
23: Per. SF hits Station 1 (data) 3: Same as 1 25: Per. SF hits Station 2 (data) 5: Same as 5

24: Per. SF hits Station 1 (MC, reco) 4: Same as 2 26: Per. SF hits Station 2 (MC, reco) 6: Same as 6

27: Per. SF hits Station 3 (data) 7:Same as 9 29: Per. SF hits Station 4 (data) 9: Same as 13

28: Per. SF hits Station 3 (MC, reco) 8:Same as 10 30: Per. SF hits Station 4 (MC, reco) 10: Same as 14

MC-Data Comparison



Percentage of MID hits

MC-Data Comparison

Results of the Kolmogorov test

HISTOGRAM ID	VARIABLE	Kolmogorov Probability
1 & 2	nsfhitrec	0.000000
3 & 4	**pulse_hgt**	**0.072083**
5 & 6	**ndchitrec**	**0.001431**
7 & 8	**emtotreco**	**0.819288**
9 & 10	nmidhitrec	0.000000
11 & 12	nmidhitrec_sc	0.000000
13 & 14	**nclu**	**0.012717**
15 & 16	**avene**	**0.648560**
17 & 18	ntksf	0.000000
19 & 20	ntkdc	0.000002
21 & 22	**ntkfin**	**0.004849**
23 & 24	nsfh_st1	0.000000
25 & 26	nsfh_st2	0.000000
27 & 28	**nsfh_st3**	**0.004136**
29 & 30	nsfh_st4	0.000000
31 & 32	nmd_st1	0.000000
33 & 34	nmd_st2	0.000000
35 & 36	nmd_st3	0.000000

- **Prob > 0.05**
- **Prob > 0.01**
- **Prob > 0.001**
- **Prob < 0.001**

Selection of Variables

- The variables we used in the neural networks are :

nsfhitrec (Number of SF hits)

ndchitrec (Number of DC hits)

ntkfin (Number of "final" tracks)

emtotreco (Total EMCAL energy)

nmd_st1 (Percentage of MID hits in WALL A)

nmd_st2 (Percentage of MID hits in WALL B)

nmd_st3 (Percentage of MID hits in WALL C)

avene (Average Cluster Energy)

nclu (Number of EMCAL clusters)

ntkdc (Number of DC tracks)

- We ended up with these particular variables (in this first approach) after several trials with different ANN structures until we obtained the best results.

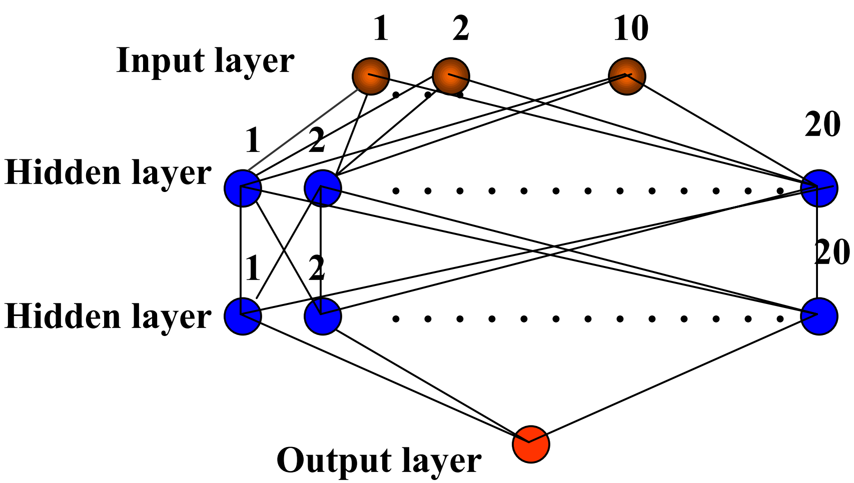
Procedure

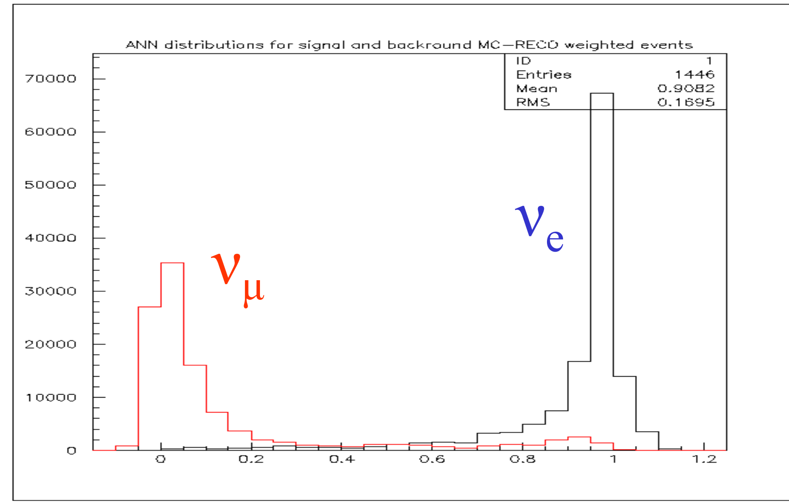
- Used **MLPfit** package through **PAW** interface to:
  - **Define** the **network structure, learning method** and **training set** (from Ntuple previous variables) both for signal and background events.
  - **Train** the network.
  - **Save** the network **results**.
- Ntuples produced by processing MC events.
- The previous procedure has been repeated many times for different networks structures, learning methods and input variables.
- **Training sets** :

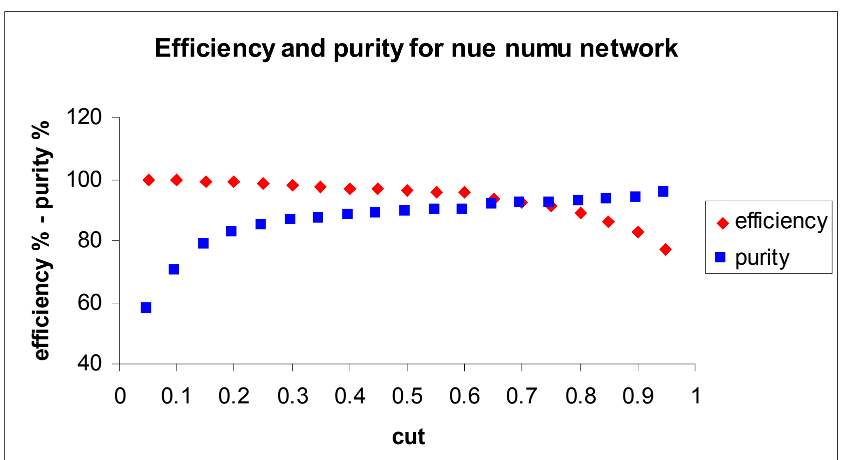
$$\underbrace{\sim\ 1500\ \nu_e - \bar{\nu}_e \quad \sim\ 1500\ \nu_\mu - \bar{\nu}_\mu \quad \sim\ 1500\ \nu_\tau - \bar{\nu}_\tau}_{\textbf{Period 4 (Interaction in module 4)}}$$

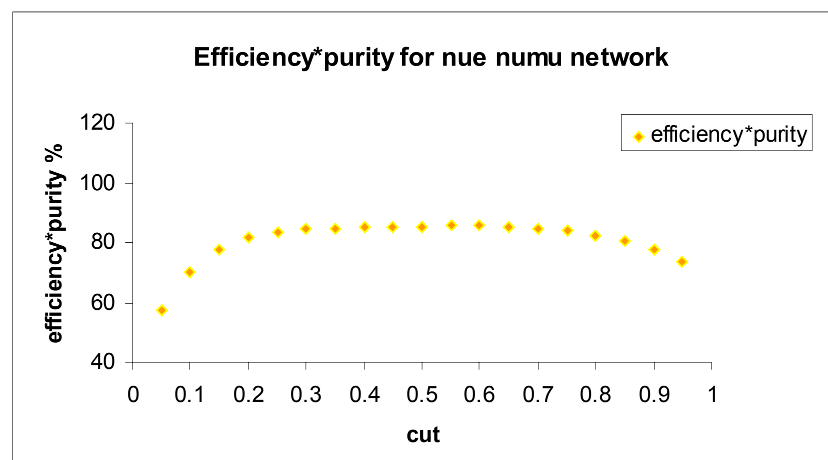
$$\underbrace{\sim\ 600\ \mathrm{CC} \quad \sim\ 600\ \mathrm{NC}}_{\textbf{Period 4 (Interaction in all modules)}}$$

Preliminary Results: ν_e-ν_μ

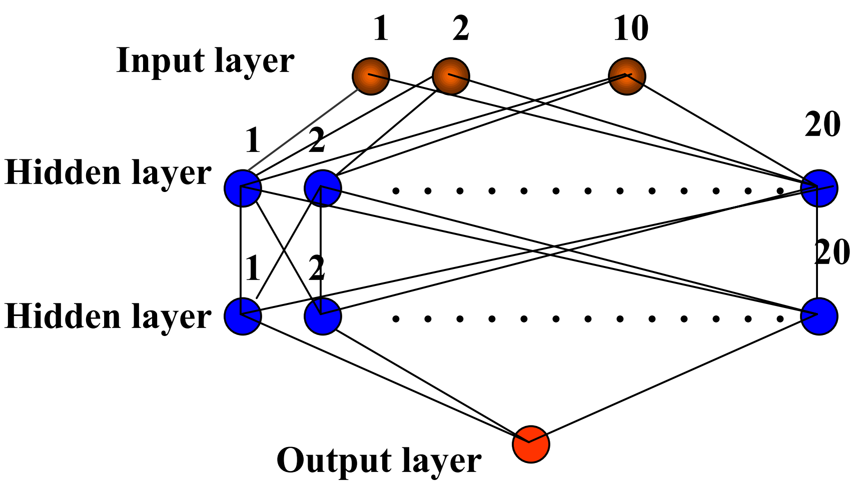


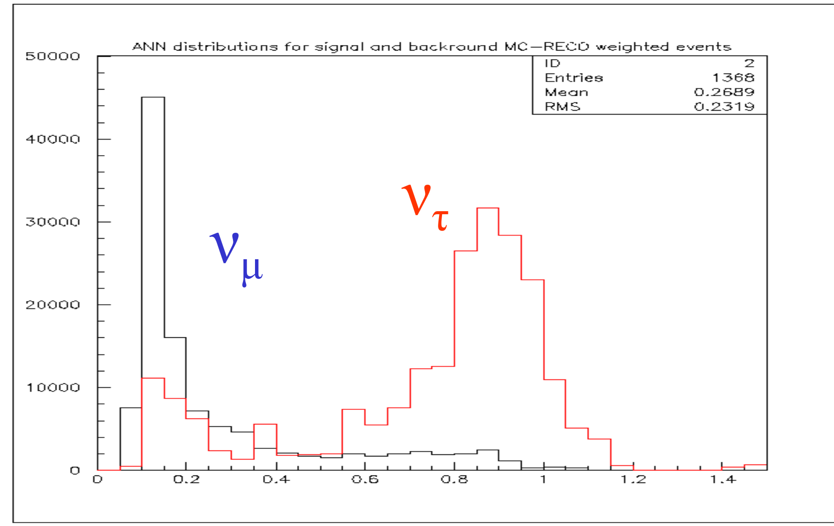


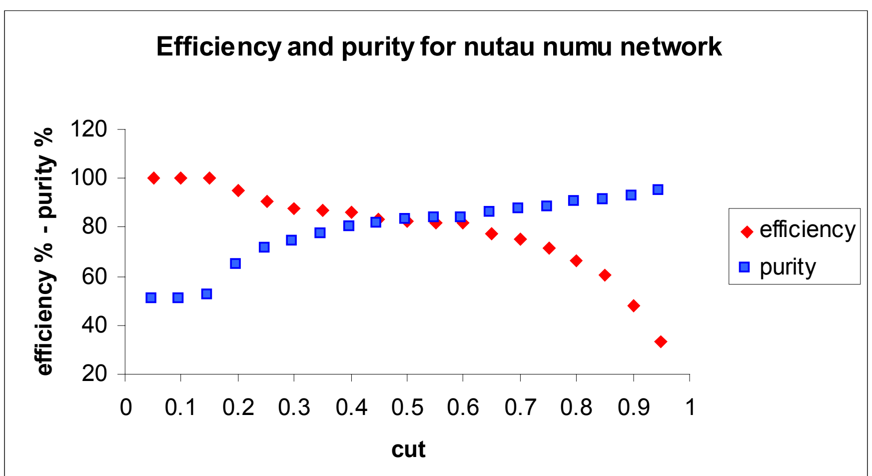


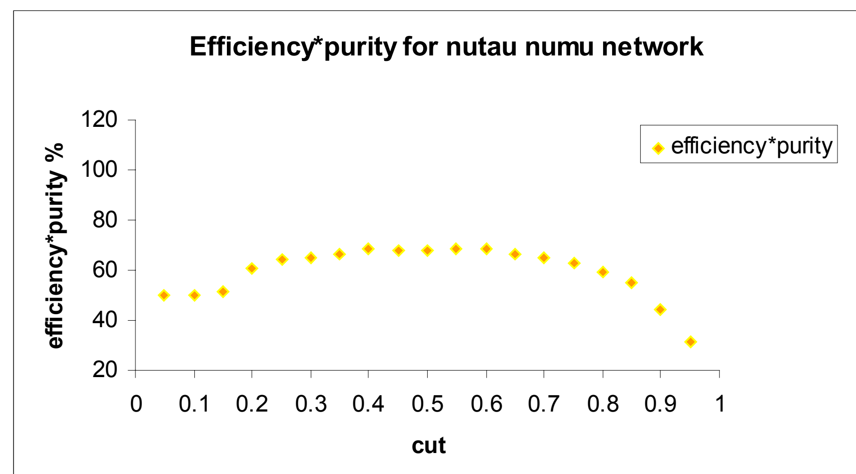


Preliminary Results: ν_τ-ν_μ

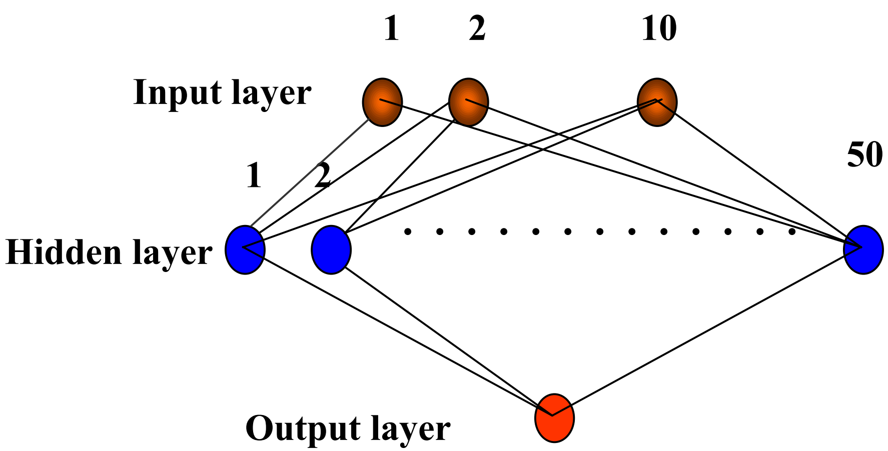


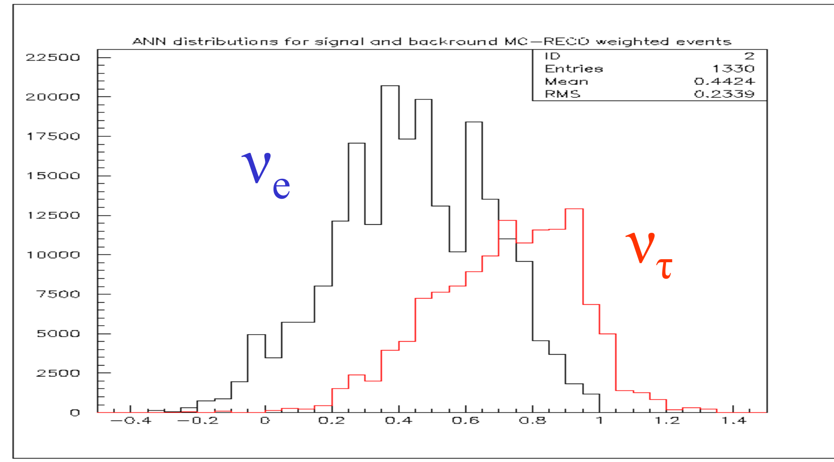


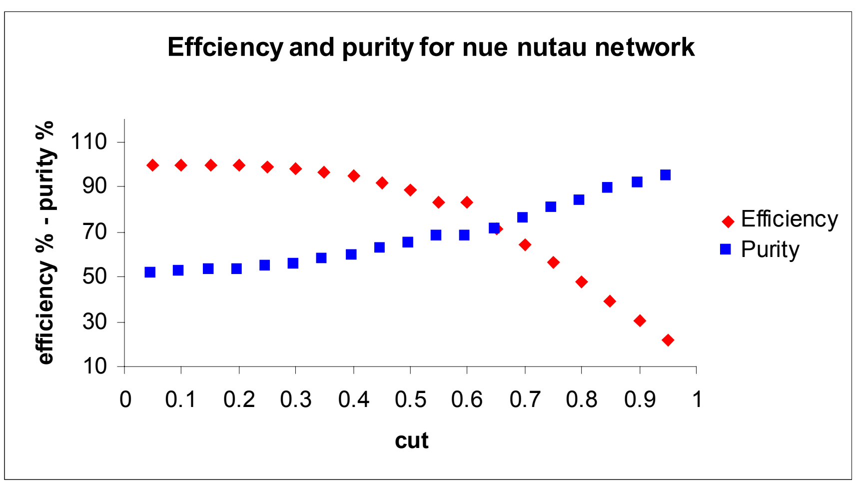


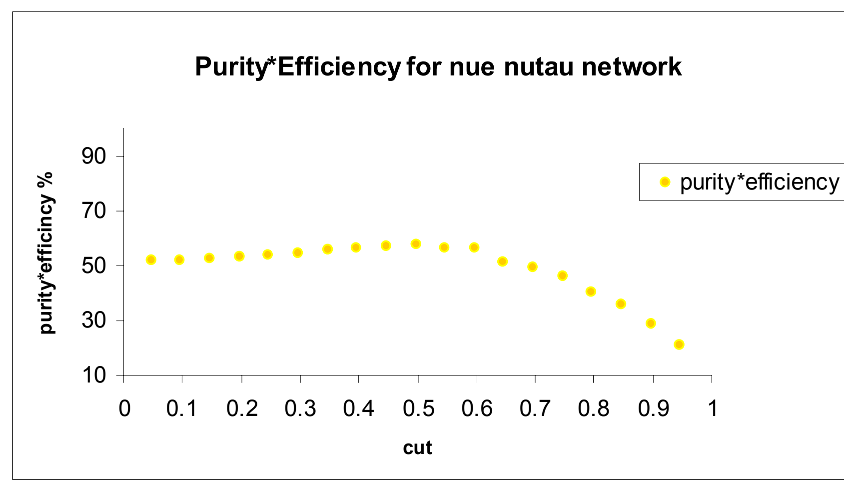


Preliminary Results: ν_τ-ν_e

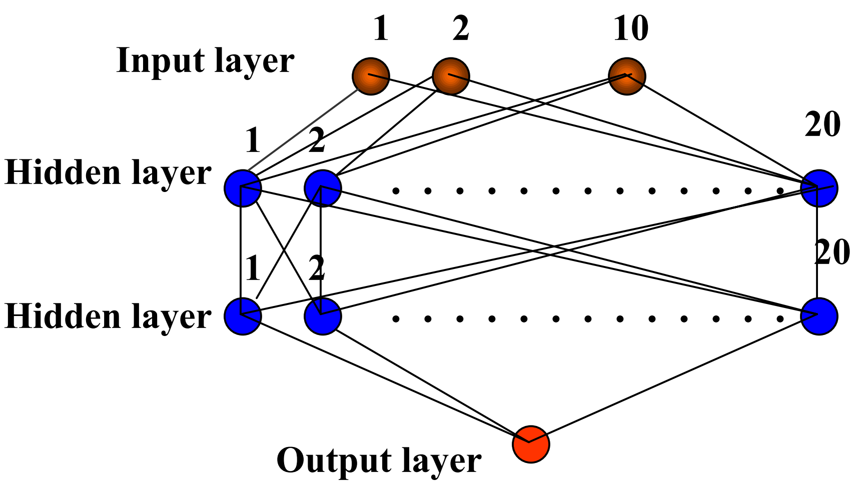


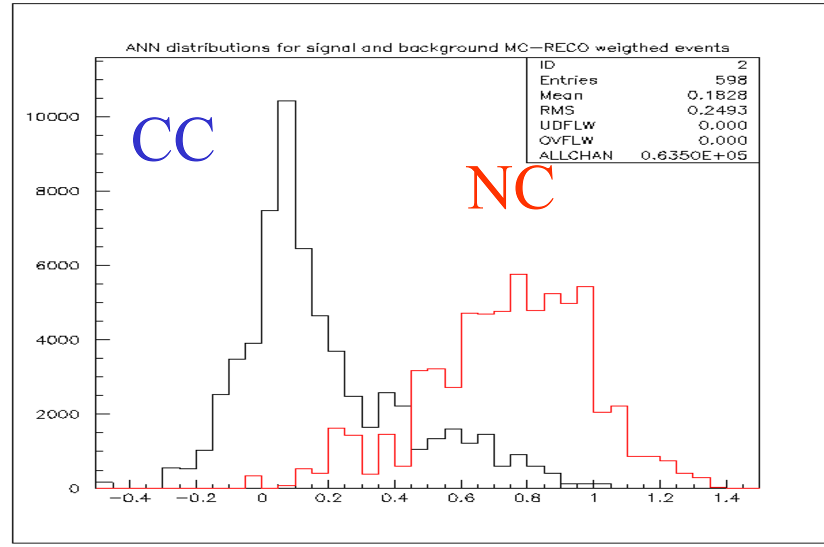


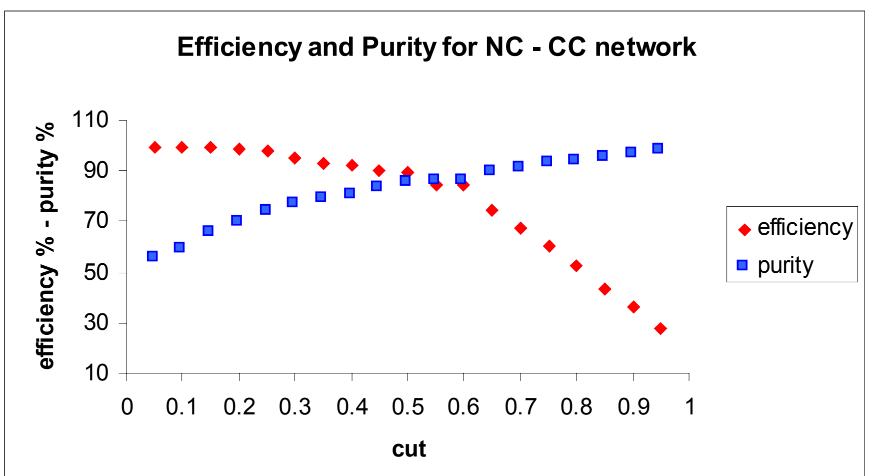


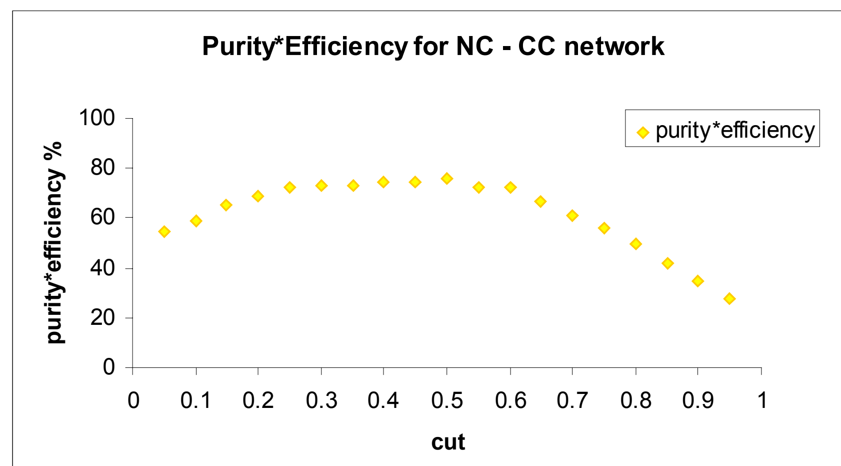


Preliminary Results: NC-CC









SUMMARY

- **Efficiency ~ 90 %** (Good Statistics but relatively poor purity)

case	per	cut	efficiency(%)	Purity(%)	eff*pur(%)
ν_e-ν_μ	4	0.80	89.0	92.3	82.1
ν_τ-ν_μ	4	0.25	90.7	70.7	64.1
ν_e-ν_τ	4	0.45	91.6	61.7	56.5
NC-CC	4	0.50	89.1	85.5	76.2

- **Purity ~ 90 %** (Poor Statistics but quite "clear " sample)

case	per	cut	efficiency(%)	Purity(%)	eff*pur(%)
ν_e-ν_μ	4	0.60	95.7	89.7	85.8
ν_τ-ν_μ	4	0.80	66.0	89.5	59.0
ν_e-ν_τ	4	0.90	30.7	91.0	27.9
NC-CC	4	0.65	74.9	89.5	67.2

CONCLUSIONS

- Employing ANN technique to do ν-event classification
- Studied various discriminating variables
- MC and data do not agree mostly on SF and MID syst
- ANN classification of ν_e-ν_μ , ν_τ-ν_μ , ν_e-ν_τ , in per 4, stat 4
- ANN classification of CC-NC in per 4, all stations

CONCLUSIONS

- The **preliminary ANN results** so far are **quite promising** and allows us to say that this **approach** can have **satisfactory** results on **event classification** (particle identification under study).

- The preliminary ANN results show that in order to successfully **complete** this **analysis** we need to create **additional ANN input variables** related with **Emulsion info.**

- The E872 **Monte Carlo** is **describing E872 data** in an **acceptable way apart** from the **SF** and **MID** system which need to be improved.

ONGOING WORK

- **Apply to 203 events**
- **Create additional ANN input variables related with emulsion info, namely :**
  - **polar angle between lepton and hadron jet**
  - **lepton angle emission with respect to the neutrino**
  - **possible kinks**
  - **daughter info e.t.c**
- **Create ANN with multivariable output, that is one ANN that will do classification in three categories (ν_e - ν_μ - ν_τ)**
- **Create ANN that will perform particle identification.**